Rule CIC167: There may be too few VSAM files assigned to LSR Pools

Finding: CPExpert has detected that there may be too few CICS VSAM files assigned to a Local Shared Resources (LSR) pool.

Impact: This finding should normally have a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: VSAM files can be assigned as nonshared resources (NSR) files or assigned to a local shared resources (LSR) pool. The major difference between the two methods is how the VSAM strings and VSAM buffers are allocated and used.

When CICS VSAM files are assigned as NSR files, there is no sharing of VSAM strings or VSAM buffers among files.

- Strings are assigned to files using the STRNO operand of the DFHFCT TYPE=FILE macro. The number of assigned strings controls the number of concurrent I/O requests to the file.

- Buffers are assigned to files using the BUFND operand (for data buffers) and BUFNI operand (for index buffers) of the DFHFCT TYPE=FILE macro. The number of assigned buffers for data records and for index records must be at least one more than the number of strings. Additional data buffers may be assigned (usually this is done to speed up VSAM CA splits by permitting chained I/O operations). Additional index buffers may be assigned and will be shared between the strings (the extra buffers can be used to hold high-level index records and thus save physical I/O).

When CICS VSAM files are assigned to LSR pools, the files share common strings and buffers assigned to the LSR pool. There is no "preallocation" of strings or buffers to particular files. Since the strings and buffers are shared, significantly fewer strings and buffers normally are required to support I/O access operations. This is because not all files will be accessed at any particular time. Please refer to Rule CIC165 for more discussion of this issue.

- Strings are assigned to LSR pools using the Resource Definition Online (RDO) **LSRPOOL** definition, or using the STRNO operand of the DFHFCT TYPE=SHRCTL macro. Alternatively, the number of

strings assigned to the LSR pool can be automatically computed by CICS using an algorithm based on the number of strings associated with files assigned to the pool. The number of assigned strings controls the number of concurrent I/O requests to the LSR pool. This number is normally much less than the sum of the strings associated with files assigned to the LSR pool (The default computed value is 50% of the number of strings assigned to files, with adjustments to ensure that the minimum required strings are assigned.).

- Buffers are assigned to LSR pools using the BUFFERS operand of the DFHFCT TYPE=SHRCTL macro. Alternatively, the number of buffers assigned to the LSR pool can be automatically computed by CICS using an algorithm based on the number of strings associated with files assigned to the pool. This number is normally much less than the sum of the strings associated with files assigned to the LSR pool (The default computed value is 50% of the number of strings assigned to files, with adjustments to ensure that the minimum required buffers are assigned.). Buffers are normally specified with different sizes, with the different sizes corresponding to the size of the control interval (CI) for data and index records of files assigned to the LSR pool.

There are some advantages to assigning files as NSR files:

- NSR allows multiple copies of a control interval (CI) in storage. An I/O request associated with one string can be updating a CI, while I/O requests associated with other strings are reading different copies of the same CI. This "read while update" is generally considered to be a poor practice, and many installations implement standards to prevent such situations.

- NSR provides better performance for VSAM CA splits if extra data buffers are allocated to the file.

- NSR allows allocation of buffers to a specific file. This advantage may be important in some unique situations with critical files. However, this advantage is normally not applicable, since critical files can be assigned to their own LSR pool and benefit from the improved "look-aside" logic implemented with LSR pools (see the discussion below).

- NSR can sometimes provide slightly better performance for sequential operations if additional buffers are allocated. This is because the processing overhead required to implement the "look-aside" logic implemented with LSR pools is not required.

There are major advantages to assigning VSAM files to LSR pools:

- If VSAM files are assigned to LSR pools, VSAM will use its "look-aside" logic to determine whether a required CI is already in a buffer, before executing any physical I/O operations. If the required record is already in a buffer, VSAM will use the record in storage, rather than issuing a read to DASD. This has the effect of implementing an in-storage caching of the file, and can **significantly** reduce the number of physical I/O operations required.

  With sufficient buffer allocation, VSAM often can find 80%-95% of the I/O requests for index records in buffers, and over 50% of the I/O requests for data records in buffers. Whether records are in a buffer is, of course, a function of the file size, the file accessing characteristics, etc. However, allocating sufficient buffers to LSR pools (and assigning VSAM files to the LSR pools) often can produce a **significant** performance improvement for CICS.

- VSAM files assigned to a LSR pool share common strings and common buffers assigned to the LSR pool. Since the strings and buffers are shared, significantly fewer strings and buffers normally are required to support I/O access operations. This is because not all files will be accessed at any particular time. Rather, file accesses will tend to be distributed across files at different times.

  Some files will have requirements for strings and buffers at one time, while at another time they will not be accessed and will not require the strings or buffers. The demand for strings and buffers therefore is the **peak collective demand** rather than the **sum** of the **peak individual** demands.

  For example, 3 files might individually have a peak I/O access demand for 5 buffers. The sum of the individual buffers required to prevent buffer waits would be a total of 15 buffers (3 * 5). However, the peak collective demand would normally be less than 15 buffers. If there were no overlap of I/O access operations among the files, the peak collective demand would be only 5 buffers.

  In practice, the peak collective demand for buffers is usually less than half of the sum of the peak individual demands. Assigning files to LSR pools therefore significantly decreases the storage requirements to support CICS VSAM buffers.

  With CICS Version 1.7, the LSR buffers moved above the 16 megabyte line. Consequently, the storage savings generally are not as important as they were with previous versions of CICS.

- Assigning files to LSR pools provides better read integrity, since there is only one copy of a CI in storage. LSR permits several read

operations to share access to the same buffer. However, updates require the exclusive use of the buffer. No other I/O request (e.g., a read or another update) is allowed access to the buffer until the update releases the buffer.

CPExpert analyzes the number of I/O requests for LSR files and computes this as a percent of all I/O requests for all VSAM files. CPExpert produces Rule CIC167 if this percent is less than the LSRIOREQ guidance variable.

The default of the LSRIOREQ guidance variable is 75, indicating that 75% of the I/O requests should be satisfied from files assigned to LSR pools. This default is set such that Rule CIC167 probably will initially be produced for many CICS regions. Many installations have allocated a relatively small number of files to LSR pools. The point of firing Rule CIC167 is to alert you to the significant performance advantages of using LSR pools. If you are not inclined to assign more VSAM files to LSR pools, change the guidance variable and prevent the firing of this rule.

Suggestion: CPExpert suggests that you assign more VSAM files to LSR pools. **The advantages of using LSR pools are so significant that VSAM files generally should be assigned to LSR pools.** There are some exceptions to this general statement:

- The file undergoes frequent VSAM control area (CA) splits.
- Files might be assigned as NSR files if there is no opportunity for the LSR "look-aside" logic to provide performance benefits. This situation could arise if the file is very active, and is a large sequentially-processed files.
- High performance is required for a critical file and the processing is unique in some way that would benefit from NSR. This exception is normally not applicable, since critical files can be assigned to their own LSR pool and benefit from the improved "look-aside" logic implemented with LSR pools.
- The processing characteristics of a particular file would dominate a LSR pool. In this case, the file probably would benefit more by assigning it to its own LSR pool, rather than assigning it as a NSR file.

However, unless you have a unique situation, you should normally assign most of your CICS VSAM files to LSR pools.

Additionally, you may wish to review the discussion in Rule CIC160 through Rule CIC166, and Rule CIC168. These rules provide insight into some of the implications of various implementation strategies.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: pages 65-68, pages 232-238, and page 244.

CICS/MVS Version 2.1.2 Performance Guide: pages 158-162, page 170, and pages 394-397.

CICS/ESA Version 3.1.1 Performance Guide: pages 71-73, pages 93-106, and page 239.

CICS/ESA Version 3.2.1 Performance Guide: pages 147-152, page 155, and pages 310-321.

CICS/ESA Version 3.3.1 Performance Guide: pages 157-162, pages 165-166, and pages 329-339.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.4.2, Section 4.4.4, and Appendix A.1.11.

CICS/TS Release 1.1 Performance Guide: Section 4.4.2, Section 4.4.4, and Appendix 1.1.9.

CICS/TS Release 1.2 Performance Guide: Section 4.4.2, Section 4.4.4, and Appendix 1.1.10.

CICS/TS Release 1.3 Performance Guide: Section 4.6.2, Section 4.6.4, and Appendix 1.1.11.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 18 (VSAM resource usage (LSRPOOL)), Chapter 18 (VSAM buffer allocations for LSR), and Appendix A (Table 53).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.5.2 Defining VSAM resource usage, Section 4.5.4 Defining VSAM buffer allocations for LSR, and Appendix 1.1.17.6.

